

Amendment to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended): A distributed file system comprising:
a storage device for holding files;
multiple clients for carrying out file operations on said storage device;
a server using tokens to control rights to file reading and writing operations by the clients; and

a network ~~connecting~~coupling said clients, said storage device and said server,

wherein said server contains a token revoke request means for sending a token revoke request for demanding a return of a token granting rights to write on said file, to a first client that holds said token,

wherein said token revoke request means sends said token revoke request, which includes information identifying a second client that requested said file, and information indicating a level of said token requested by said second client, said level being either read or write, and

wherein said first client comprises a memory section for holding file data loaded from said storage device and a data output means for sending said file held in said memory section and relating to said token, to said server for said second client that requested said token when said token revoke request is received, and

wherein said data output means sends data in a range among files linked by said token, to said server for said second client that requested said token, and performs synchronous processing on said storage device by writing data in a range among said files not linked by said token.

2. (canceled).

3. (previously presented) The distributed file system according to claim 1, wherein the file relating to said token sent from said first client to the server for said second client that requested said token, contains information not already appearing in the latest information in said storage device.

4. (canceled).

5. (previously presented): The distributed file system according to claim 1, wherein said data output means decides whether to send said token of said file held in said memory section to said server for the second client that requested the token, or to write said file in said storage device and perform synchronous processing on said storage device, based on an input/output capacity of said network and said storage device and/or a data size of said file sent to said server for said second client that requested the token.

6. (currently amended): A file send and receive method utilized in a distributed file system, wherein said distributed file system includes a storage device

for holding files, multiple clients for carrying out file operations on said storage device, a server using tokens to control rights to file reading and writing operations by the clients, and a network ~~connecting~~coupling said clients, said storage device and said server, said method comprising:

making a request to said server for a token for rights to perform said file operations, wherein a first client makes the request to said server;

sending, by said server, said token revoke request to a second client that holds write operation rights to said file, so as to request a return of the token for said write operation rights,

wherein said token revoke request includes information identifying the first client that requested the token for said file, and information indicating a level of the token requested by said first client, said level being either read or write; and

sending, by said second client that received said token revoke request, the file for said token held in said memory section, to the first client that requested the token for said file,

wherein said first client that received the token revoke request sends data in a range among files linked by said token to the server for said second client that requested said token, and performs synchronous processing on said storage device by writing data in a range among files not linked by said token on said storage device.

7. (canceled).

8. (previously presented): The file send and receive method according to claim 6, wherein the file relating to said token sent from said first client that received the token revoke request to said server for said second client that requested said file does not show the latest information in said storage device.

9. (canceled).

10. (previously presented): The file send and receive method according to claim 6, wherein said first client that received the token revoke request, decides whether to send the file being held to the server for the second client making the request for said token, or to write the file in said storage device and perform synchronous processing of said storage device, based on an input/output capacity of said network and said storage device, and/or a data size of the file sent to the server for the second client that requested the token.

11. (currently amended): A first client device utilized in a distributed file system, wherein said distributed file system includes a storage device for holding files, multiple client devices for carrying out file operations on said storage device, a server using tokens to control rights to file reading and writing operations by the client devices, and a network ~~connecting~~coupling said clients devices, said storage device and said server, said first client device comprising:

a memory section for holding file data loaded from said storage device; and

a data output means for sending a file relating to a token held in said memory section to a second client device that requested the token for said file when a

request for returning said token for rights to read or write on said file is received by said first client device from said server,

wherein said request includes information identifying said second client that requested said token for said file, and information indicating a level of said token requested by said second client, said level being either read or write, and

wherein said data output means sends data in a range among said files linked by said token, to said server for said second client that requested said token, and performs synchronous processing on said storage device by writing data in a range among files not linked by said token.

12. (previously presented): The first client device according to claim 11, wherein the file relating to said token sent to said server for the second client device that requested said token, does not show the latest information on said storage device.

13. (canceled).

14. (previously presented): The first client device according to claim 11, wherein said data output means decides whether to send the file being held to the server for the second client that requested said token, or to write the file in said storage device and perform synchronous processing of said storage device, based on an input/output capacity of said network and said storage device and/or a data size of the file sent to the server for the second client that requested the token.

15. (currently amended): A program embedded on a computer-readable storage medium, wherein the program is executed on a server device for controlling tokens for rights to file reading and writing by a first client ~~connected~~coupled via a storage device and network, wherein said first client requests a token for rights to file reading or writing, and wherein said program causes the server device to:

send a request for a return of said token for rights to file reading or writing, to a second client that holds rights to read or write on a file,

wherein said request includes information identifying said first client that requested said token for said file, and information indicating a level of said token requested by said request client, said level being either read or write, and

wherein said first client that received the token revoke request sends data in a range among files linked by said token to the server for said second client that requested said token, and performs synchronous processing on said storage device by writing data in a range among files not linked by said token on said storage device.

16. (currently amended): A program embedded on a computer-readable storage medium, wherein the program is executed on a first client device for controlling rights to reading and writing of files stored on a storage device ~~connected~~coupled by a network, by utilizing tokens managed by a server, wherein said first client requests a token for the rights to reading and writing, and wherein said program causes the first client device to:

send a file relating to the token held in said storage section to a second client device that requested said token for said file, when a request to revoke said token for rights to write on said file is sent from said server,

wherein said request includes information identifying said second client that requested said token for said file, and information indicating a level of said token requested by said second client, said level being either read or write, and

wherein said first client that received the token revoke request sends data in a range among files linked by said token to the server for said second client that requested said token, and performs synchronous processing on said storage device by writing data in a range among files not linked by said token on said storage device.

17. (new): The program according to claim 15, wherein said first client that received the token revoke request, decides whether to send the file being held to the server for the second client making the request for said token, or to write the file in said storage device and perform synchronous processing of said storage device, based on an input/output capacity of said network and said storage device, and/or a data size of the file sent to the server for the second client that requested the token.

18. (new): The program according to claim 16, wherein said first client that received the token revoke request, decides whether to send the file being held to the server for the second client making the request for said token, or to write the file in said storage device and perform synchronous processing of said storage device,

based on an input/output capacity of said network and said storage device, and/or a data size of the file sent to the server for the second client that requested the token.